

Density-Aware Selection Tools

Paymahn Moghadasian
University of Manitoba
66 Chancellors Cir
umpaymah@cs.umanitoba.ca
?

Pourang Irani
University of Manitoba
66 Chancellors Cir
irani@cs.umanitoba.ca
?

Niklas Elmqvist
Purdue University
?
elm@purdue.edu
?

ABSTRACT

In this paper we present selection tools that utilize pre-existing knowledge of the distribution of the underlying data to ease browsing and selection. These new data-aware selection tools include a slider, a range slider, and an improved lasso selection method. Standard models of these tools suffer from two common problems: poor subpixel data querying, and uniform visual representation of non-uniform data. We provide density-aware interactions to give users more efficient subpixel data querying and use embedded visualisations to better represent the data encoded by the tools. Through several controlled user studies, we find that our proposed density-aware slider outperforms standard alphasliders and default trackbars. We also find that a density-aware range slider and lasso selection outperform current state-of-the-art methods.

Author Keywords

Dynamic query, information visualization, slider, data selection, Alphaslider, lasso, range slider

ACM Classification Keywords

Need help with this

General Terms

Need help with this

INTRODUCTION

Dynamic queries provide easy to use, powerful and efficient tools and interfaces which allow users to rapidly and reversibly query data and uncover trends in the data being explored [1]. Currently, there are many tools available to users which benefit from the advantages of dynamic queries such as the slider, checkbox or button. The Alphaslider is a slider designed to query large lists of alphabetically sorted alphanumeric items [2]. The range slider is a slider which allows users to query a range of data items. This is a natural extension of the Alphaslider which only allows for single item querying. The lasso, a technique used to select multidimensional data, is an extension of the range slider where users lasso graphed or visualized data to highlight or zoom in on it.

The Alphaslider, range slider and the lasso technique suffer from two common problems. None of these tools have an awareness of the data they encode. This is especially important when the interfaces map more than one item to any pixel. Subpixel querying is a new problem because information density is ever increasing. With highly dense data more items are mapped to each pixel making efficient data selection increasingly difficult. The second problem is how each tool visualizes data in general and in high density areas. Many times there is no way for the user to know whether a given pixels maps to few or many items.

We propose density-aware selection tools. These tools are knowledgeable of the data they encode and use this information to simplify data querying. Our most efficient slider interface intelligently redistributes the items in densely packed pixels over a larger area and provides users with a list of nearby items giving the user a sense of location within the pixel. <Summarize range slider and lasso>. To solve the issue of data visualisation we used embedded visualisations such as those proposed by Willett et al [3]. These visualizations aid in estimation of the density of items in a given pixel.

Our results show that users can query significantly faster with data aware widgets when compared against the standard tools. **Interface X** allowed subjects to query significantly faster than both the Alphaslider and the default trackbar. Our proposed range slider outperforms standard range sliders while our lasso technique gave users no improvement for low and medium density information but showed large improvement over the current lasso technique for high density information.

This paper provides novel solutions to an aging problem and demonstrates that intelligent selection tools have the potential to improve upon the status quo. Although the proposed solutions perform more efficiently than previous standards there is still work to do in fine-tuning and possibly developing hybrid designs that take advantage of aged and novel ideas. In this paper we discuss will present the problem in greater detail, discuss related works and continue to describe the experimental design and results of our controlled user study. We will conclude the paper by discussing the implications of the research.

RELATED WORK

Need to write this

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

EXPERIMENTAL DESIGN

Introduction

An experiment was conducted to compare different designs of density aware sliders against the Alphaslider and the standard Trackbar found in WinForms.

Apparatus

The interfaces used in the experiment were built using Visual Studio 2010 Professional. A 23.5 inch Dell monitor with a resolution of 1920x1080 pixels with a standard 3 button mouse was used. Query results were displayed in Microsoft Sans Serif with a font size of 8.25 while query targets were displayed in Microsoft Sans Serif with a font size of twelve. An Intel i5-2400 CPU with a clock speed of 3.1 GHz was used along with 4 GB of RAM.

Interfaces

A total of 7 different sliders were used in the experiment (Figures 1 through 6). In all cases the text output was RSVP and was displayed under the slider. Between searches subjects had to press a “Start” button indicating they were ready and upon completion of the search the subject had to press a “Confirm” button indicating they have completed the search. **This is going to be changed. Instead of clicking on buttons the user will toggle states by pressing the space bar.**

Windows Trackbar

The Trackbar (Figure 2) is what software designers using Visual Studio find as the default slider. With this slider users can navigate through the list by dragging the slider thumb to various pixels. Clicking to the left or right of the thumb will navigate five items behind or ahead of the current one while using the left and right arrows on the keyboard will navigate one item at a time.

Alphaslider

The Alphaslider (Figure 1) has four navigation techniques. Users can jump directly to an item in the list by clicking anywhere in the bounded area above the letters. Users can navigate through the list at a rate of ten items per mouse movement by clicking in the top tile of the slider thumb and dragging. Users can also navigate one item at a time either by clicking on the arrows at the ends of the slider or by clicking in the bottom tile of the slider thumb and dragging.

ActiveArea Slider

The ActiveArea Slider (Figure 3) allows users to navigate by clicking on and dragging the slider thumb, by clicking on and dragging the secondary red slider or by pressing the left or right arrow keys on the keyboard. The main slider has a variable size which changes based on the density of information of the pixel it queries. Dragging the secondary red slider allows users to query other items found in the pixel being queried by main slider. The red slider is bound by the edges of the main slider.

Histogram Slider

The Histogram Slider (Figure 4) functions very similarly to the ActiveArea Slider. The most significant difference is

that the secondary red slider moves up and down along the histogram as opposed to left and right along the slider. The size of the slider stays constant. Users can still navigate one item at a time by pressing the left and right arrow keys.

MouseWheel Slider

The MouseWheel Slider (Figure 5) is also quite functionally similar to the ActiveArea slider. However, this slider allows subpixel querying through the use of the mouse wheel. Rolling the mouse wheel up will cause the next item in the list to be queried while rolling the mouse wheel down has the opposite effect.

MultiValue Slider

Need to do pilots and find out which (if either) multivalue slider one is better

Hypotheses

This paper is primarily concerned with designing the most efficient slider. While accuracy is an important factor in designing a slider that attribute falls largely upon the user. Because of this speed is the most significant factor to measure. The speed with which a user locates an item is largely dependent on total mouse movement. For each of the sliders there is a period of querying where the user approximates the area of the target and a period of querying where the user does fine adjustments to acquire the target. Based on this assumption the following hypotheses can be made:

- 1) The MultiValue slider will outperform all other sliders in low local density areas because additional mouse movement will be minimized.
- 2) The Histogram Slider will be faster than the ActiveArea Slider because reacquiring the ActiveArea Slider's slider in low local density areas is difficult.
- 3) The MouseWheel Slider will perform better than the Alphaslider because rolling the mouse wheel is far more precise than moving the mouse.
- 4) The Trackbar will perform the worst because it depends on keyboard input for single item navigation.

Experiment Variables

Independent Variables

- i) Type of interface
- ii) Target density
- iii) Data size

Dependent Variables

- i) Speed of acquisition
- ii) Error rate
- iii) Subjective satisfaction

Tasks – rewrite this part

Subjects completed three trials for each permutation of the independent variables. Each target was randomly generated at runtime. To mitigate learning effects new data was randomly selected upon interface change. With each new task the thumb was brought to the beginning of the slider.

Participants**Procedures**

A pilot study was first done to weed out poorly designed and inefficient interfaces. Prior to the timed trials the subjects were given five minutes to familiarize themselves with each interface while reading interface specific instructions and ask questions. During the timed trial the subjects were not allowed to ask questions. Upon completion of the experiment the subjects were asked to fill out a questionnaire.

RESULTS

Need to write this

CONCLUSION

Need to write this

ACKNOWLEDGMENTS

Need to write this

REFERENCES

- [1] Shneiderman B., "Dynamic Queries for Visual Information Seeking," Maryland, 1994.
- [2] Christopher Ahlberg and Ben Shneiderman, "The Alphaslider: A Compact and Rapid Selector," in *Human Factors in Computing Systems*, Boston, 1994, pp. 365-371.
- [3] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala, "Scented Widgets: Improving Navigation Cues with Embedded Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129-1136, November 2007.

The columns on the last page should be of approximately equal length.

Remove these two lines from your final version.